Css

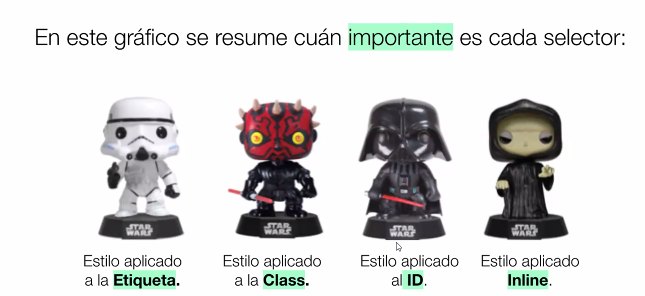
/\* comentario \*/

Para agregar css a html, es yendo a head y poner link css

Style=”width:x”

Class: es para elegir el nombre para poder modificarlo en css, ya q si ponemos h1{ color: red; } se nos van a poner todos los h1 de rojo

Para llamar a una clase se pone un . ej: .textosResaltados{color: blue;} entonces solo donde pusimos esta clase se va a colorear de azul



Jerarquías dentro de css

class=””: etiqueta de clase. Se usa para editar con css a lo que nosotros digamos. A la clase se la llama con un ‘.’

ID es un selector, no puede existir ningún id repetido. Se lo llama con #. Se lo usa mas para hacer rutas a secciones de nuestra pagina, para q se scrolee a donde digamos.

Ul li a{ color: purple} esto significa q a todos lo a dentro de un li que estén dentro de un ul van a ser violeta, podes ponerle una clase al principio para ser mas específicos.

Tabla de puntajes: etiqueta vale 1, class vale 10, id vale 100 e inline vale 1000

Text-align<> es para alinear los textos

Line-height<> altura entre letras

Letter-spacing<> espacio entre letras

Text-decoration<> modificar texto

List-style: none para sacarle los puntos al ancor

Background-image: url (“”) Para poner una foto de fondo

Background-position: para poner la foto donde querramos

Background-size: para el tamaño

Height: para poner la altura del body, header etc

BEM

Bloque (padre)

Elemento (los descendientes) es con \_\_

Modificador: es con --, se usa mas q nada para resaltar un link cuando tenemos el mouse arriba

**Box modeling**

Los porcentajes son en relación a sus papas, no a la pantalla

Con width fit content hace que el botón no se corte cuando achicamos la pantalla

Width: 50vw no tiene en cuenta al padre como el % sino que toma en cuenta el ancho total de la pantalla del usuario

Height: 50vh toma en cuenta la altura total de la pantalla del usuario

Border: solid, dash, etc (tipo) (x)px y el color

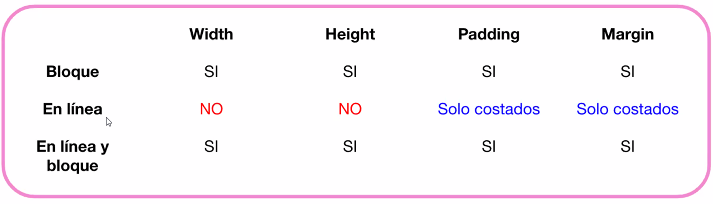
Border radius: para ver cuan redondeado va a estar el borde

Padding: del contenido hacia el borde, puede recibir hasta 4 valores. Padding: 0 20 px

Margin: lo mismo que el padding pero para afuera del borde

Como decirle a un class que sea de bloque y no de línea:

Con Display es para ver como vamos a mostrar los elementos. Display: inline-block



Overflow: scroll es para q si el texto sobresale del borde, hace que lo ponga dentro y lo scrollee.

**Positions**

Posicionan elementos de una manera diferente de la que ya están

*Position: fixed;* se usa para que un botón siga en la pantalla aunque scrolleemos, no mantiene el espacio que ocupa, se corren los de arriba al espacio. Con fixed nos va a desaparecer el logo, muy importante ponerle la posición q queremos que este para que nos aparezca

Con un right: 15px le decimos que este a la derecha

Con un botton:15px aparecerá abajo. Con un to:15px le decimos que aparezca arriba.

*Position: sticky* no nos corrompe el nav y queda igual al que teníamos desde un principio. Se mantiene el espacio que ocupa cuando está en su lugar pero se corre cuando scrolleamos. Es ideal para el nav, la barra de navegación

Top 0: seria para cuando la parte que querramos que scrollee llegue a el top, va a empezar a bajar

*Position: Absolut* para hacerlo absoluto hay que hacer al padre *Relative,* si no lo hacemos, cuando superpongamos un texto a una imagen se va a acomodar según al body y no a la imagen.

*Menú desplegable:*

Dentro del li vamos a tener un ul

Li{

Position: relative;

Height:200px

}

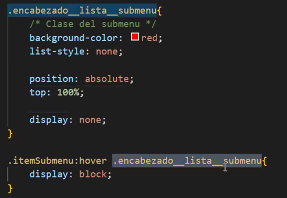
Ul{

Position: absolute (esto para que se superponga al li)

Top: 100% (esto para que se ponga en el top del li pero desde abajo, para el menu)

Display: none; (para que el menú desaparezca)

}



La segunda clase dice que cuando a itemsubmenu se le pase el mouse va a aparecer el encabezado (diplay:block)

**FLEXBOX**

Se usa para ordenar, en lugar de poner inline-block, etc. Lo primero es indicarle al padre un display: flex para que se ordenen los hijos

Flex-direction: row; nos dice cuál va a ser el eje principal y cual el secundario, esto va en segundo lugar

Para ordenar:

Justify-content: ordena según el eje principal, en este caso el row (horizontal)

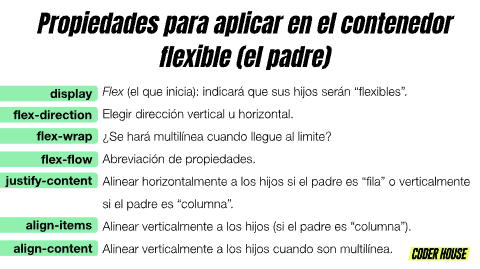
Justify-content: space-evenly ordena en partes iguales el espacio q las separa según el tamaño de pantalla

Justify-content: space-around

Align-items: nos alinea el contenido según el eje secundario

Flexwrap: hace q las imágenes se ajuste según el usuario, si achica la pantalla las imágenes se iran para abajo.

Align-content: nos alinea el contenido según el flexwrap y el alto. Solo funciona cuando tenemos el Flexwrap



Object-fit: es necesario haber puesto un height (las imágenes en las paginas no deberían pesar mas de 200kb)